A Survey of Routing Protocols for Mobile Ad-Hoc Networks with Comprehensive Study of OLSR

A. Fatemeh Sarkohaki, B. Shahram Jamali

Computer Engineering Department, Islamic Azad University, Germi Branch

Computer Engineering Department, University of Mohaghegh Ardabili

Abstract

Mobile Ad-Hoc Network (MANET) includes a collection of wireless mobile hosts forming a temporary network without the aid of any stand-alone infrastructure or centralized administration. MANET is a self-organized and self-configurable network in which the mobile nodes move randomly. In MANET's appropriate routing algorithm is necessary to find special routes between source and destination. The main goal of any ad-hoc network routing protocol is to meet the challenges from the dynamically changing topology and establish an efficient route between any two nodes with minimum routing overhead and bandwidth consumption. The protocols are different in terms of routing methodologies and the information used to make routing decisions. Therefore a several protocols are introduced for improving the routing mechanism to find route between any source and destination host across the network. In this paper we present a survey on routing protocols and we review OLSR protocol and improvements on this protocol.

Keywords: Routing Protocols, MANET, Ad-Hoc, Routing, table driven protocol, On-demand protocol, OLSR Protocol.

I. Introduction

A mobile ad hoc network (MANET) is a self-organizing and self configuring multi hops wireless network, where the network structure changes dynamically because of member mobility (Murthy et al,2004). Nodes are free to move randomly and organize themselves arbitrarily; each node will be able to communicate directly with any other node that resides within its transmission range. Thus, the network’s wireless topology may change rapidly and unpredictably (Awerbuch et al,2002) (Macker et al,1999) (Abusalah et al,2008). Such a network may operate in a standalone fashion, or may be connected to the larger Internet (Panda et al,2008) (Chiang et al,1997) (Khetarpal et al,1997). Routing in ad hoc networks has been an active research area and in recent years numerous routing protocols have been introduced for MANETs (Johnsort et al,1994)(Buchegger et al,2002)(Hauser et al, 1999)(Coppersmith et al,2002). MANET can be applied to various applications including battlefield communications, emergency relief scenarios, law enforcement, public meeting, virtual class room and other security-sensitive computing environments. There are 15 major issues and sub-issues involving in MANET such as routing, multicasting/broadcasting, location service, clustering, mobility management, TCP/UDP, IP addressing, multiple access, radio interface, bandwidth management, power management,
security, fault tolerance, QoS/multimedia, and standards/products (Tamilarasan et al, 2012). Currently, the routing, power management, bandwidth management, radio interface, and security are hot topics in MANET research. The routing protocol is necessary whenever the source needs to transmit and delivers the packets to the destination.

Routing approaches in Mobile Ad Hoc Network:

1. In ad hoc mobile networks, routes are mainly multi hop due to the limited radio propagation range and topology changes frequently and unpredictably since each network host moves randomly. Therefore, routing is an integral part of ad hoc communications.

2. Routing is to find and maintain routes between nodes in a dynamic topology with possibly uni-directional links, using minimum resources.

The rest of this paper is organized as follows. In Section 2, we present classification of existing routing protocols. In Section 3, we review olsr protocol and improvements on this protocol. Section 4 summarize the methods in the table are expressed. We conclude our discussion in Section 5.

II. Taxonomy for Routing Protocols in MANET

MANET protocols are used to create routes among multiple nodes in mobile ad-hoc networks. IETF (Internet Engineering Task Force) MANET working group is responsible to analyze the problems in the ad-hoc networks and to observe their performance. There are different reasons for designing and classifying routing protocols for wireless ad-hoc networks. The MANET protocols are classified into three huge groups, namely Proactive (Table-Driven), Reactive (On-Demand) routing protocol and hybrid routing protocols (Tamilarasan et al, 2011)(Chen et al, 1998)(Das et al, 1998)(Pei et al, 2000)(Abolhasan et al, 2004). The following figure shows the classification of protocols.

![Ad-Hoc Routing protocols](image)

Figure.1. Different type of routing protocols in wireless Ad-hoc network (Tamilarasan et al, 2012)
Proactive routing protocols attempt to sustain consistent, up-to-date routing information between every pair of nodes in the network by propagating, proactively, route updates at fixed intervals. Representative proactive protocols include: Destination-Sequenced Distance-Vector (DSDV) routing, Clustered Gateway Switch Routing (CGSR), Wireless Routing Protocol (WRP), Optimized Link State Routing (OLSR) and The Fisheye State Routing (FSR).

a. **Destination-Sequenced Distance-Vector (DSDV) routing**

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. The primary contribution of the algorithm was to solve the Routing Loop problem. DSDV works in the following way. Each routing table entry carries hop distance and next hop for all available destinations (as in B-F). In addition, each entry is tagged with a sequence number which originates from the destination station. The routing information is advertised by broadcasting periodically and incrementally. Upon receiving the routing information, routes with more recent sequence numbers are preferred as the basis for making forwarding decisions of the paths with the same sequence number; those with the shortest hop distance will be used. That information (i.e. next hop and hop distance) is entered in the routing table, along with the associated sequence number tag. When the link to the next hop has failed, any route through that next hop is instantly assigned an infinite hop distance and its sequence number is updated. When a node receives a broadcast with an infinite metric, and it has a more recent sequence number to that destination, it triggers a route update broadcast to disseminate the important news about that destination (Perkins et al., 1994; Usop et al., 2009).

**Advantage:**

- The DSDV protocol is quite suitable for creating ad hoc networks with small numbers of nodes.
- The DSDV protocol is proven to guarantee loop-free paths to each destination at all instants.
- DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle.
- DSDV is not suitable for highly dynamic networks. There is no commercial implementation of this algorithm.

b. **Optimized Link State Routing Protocol (OLSR)**

OLSR is a proactive routing protocol for mobile ad hoc networks. The protocol inherits the stability of the link state algorithm and has the advantage of having routes instantly available when needed due to its proactive nature. OLSR minimizes the overhead caused by flooding of control traffic by using only selected nodes, called Multi-Point Relays (MPR), to retransmit control messages. This technique significantly reduces the number of retransmissions required to flood a message to all nodes in the network. Upon receiving an update message, the node determines the routes (sequence of hops) toward its known nodes. Each node selects its MPRs from the set of its neighbors saved in the Neighbor list. The set covers nodes within a distance of two hops. The idea is that whenever the node broadcasts the message, only the nodes included in

OLSR uses HELLO and TC messages. The Topology Control (TC) messages for continuous maintain of the routes to all destinations in the network, the protocol is very efficient for traffic patterns where a large subset of nodes is communicating with another large subset of nodes, and where the [source, destination] pairs change over time. The HELLO messages are exchanged periodically among neighbor nodes, to detect the identity of neighbors and to signal MPR selection. The protocol is specially suited for large and dense networks, as the optimization is done by using MPRs which work well in this context. The larger and more dense a network, the more optimization can be achieved as compared to the classic link state algorithm. OLSR uses hop-by-hop routing, i.e., each node uses its local information to route packets (Akyildiz et al, 2005)(Iannone et al, 2005)(Sundaresan et al, 2004)(Tamilarasan et al, 2012).

![Fig.2: Packet Transmission Using MPR (Tamilarasan et al, 2012).](image)

**Advantages:**

- OLSR does not need central administrative system to handle its routing process.
- The link is reliable for the control messages, since the messages are sent periodically and the delivery does not have to be sequential.
- OLSR is suitable for high density networks.
- It does not allow long delays in the transmission of packets.

**B. On-demand or Reactive Protocols**

A different approach from table-driven routing is reactive or on-demand routing. Reactive protocols, unlike table-driven ones, establish a route to a destination when there is a demand for it, commonly initiated by the source node through discovery process within the network. Reactive protocols, unlike table-driven ones, establish a route to a destination when there is a demand for it, usually initiated by the source node through discovery process within the network. Representative reactive routing protocols include: Dynamic Source Routing (DSR), Ad hoc On Demand Distance Vector (AODV) routing, Temporally Ordered Routing Algorithm (TORA) and Associatively Based Routing (ABR).
a. **Dynamic Source Routing (DSR)**

Dynamic Source Routing (DSR) is a routing protocol for wireless networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. There are 2 major phases:-


The protocol allows multiple routes to any destination and allows each sender to select and control the routes used in routing its packets, for example for use in load balancing or for increased robustness. Other advantages of the DSR protocol include easily guaranteed loop-free routing, support for use within networks containing unidirectional links, use of only "soft state" in routing, and very two hundred nodes, and is designed to work well with even very high rates of mobility (Johnson et al, 2007) (Johnson et al, 2001).

![Figure 3. DSR route discovery](image)

**Advantage:**

- Route maintenance in this protocol is fast and simple.
- In case of a fatal error in the data-link layer, a route error packet is generated from a failing node.
- When the route-error packet is received, the failing node is removed from its route cache, and all routes containing that node are truncated.

b. **Ad-hoc on demand Vector Protocol (AODV)**

AODV combines some properties of both DSR and DSDV. It uses route discovery process to cope with routes on demand basis. It uses routing tables for maintaining route data. It is reactive protocol; it doesn’t need to maintain routes to nodes that are not communicating (Sarkohaki et al, 2012).

AODV handles route discovery process with Route Request (RREQ) messages. RREQ message is broadcasted to neighbor nodes. The message floods through the network until the desired destination or a node knowing fresh route is reached. Sequence numbers are used to guarantee loop freedom. RREQ message cause bypassed node to allocate route table entries for reverse route. The destination node uncast a Route Reply (RREP) back to the source node. Node transmitting a RREP message creates routing table entries for forward route (Huawei et al, 2008) (Perkins et al, 1999) (Ashtiani et al, 2010) (Gupta et al, 2010)

Figure (Fig.4) shows, AODV routing protocol with RREQ and RREP message.
For route maintenance nodes periodically send HELLO messages to neighbor nodes. If a node fails to receive three consecutive HELLO messages from a neighbor, it concludes that link to that specific node is down. A node that detects a broken link sends a Route Error (RERR) message to any upstream node. When a node receives a RERR message it will indicate a new source discovery process. Figure (Fig.5) shows AODV routing protocol with RERR message.

**Advantages:**
- Routes are established on demand and destination sequence numbers are used to find the latest route to the destination.
- Lower delay for connection setup.

**C. Hybrid Routing Protocols**

Purely proactive or purely reactive protocols perform well in a limited area of network setting. However, the diverse applications of ad hoc networks across a wide range of operational conditions and network configuration pose a challenge for a single protocol to operate efficiently. Researchers advocate that the issue of efficient operation over a wide range of conditions can be addressed best match these operational conditions (Abusalah et al, 2008).
Representative hybrid routing protocols include: Zone Routing Protocol (ZRP) and Zone-based Hierarchal Link state routing protocol (ZHLS).

a. Zone routing protocol (ZRP)
The Zone Routing Protocol (ZRP) is a hybrid routing protocol, where the network is divided into routing zones according to the distances between nodes and the routing zone defines a range (in hops) that each node is required to maintain network connectivity proactively. The proactive part of the protocol is restricted to a small neighborhood of a node and the reactive part is used for routing across the network. This reduces latency in route discovery and routing zone is k, each node in the zone can be reached within k hops from S. The minimum distance of a peripheral node from S is k (the radius). All nodes except L are in the routing zone of S with radius 2 (Panda et al, 2012)(Tamilarasan et al, 2012).

In this proactive routing approach—Intra Zone Routing Protocol (IARP) is used inside routing zones and reactive routing approach—Inter Zone Routing Protocol (IERP) is used between routing zones. Therefore, for nodes within the routing zone, routes are immediately available. For nodes that lie outside the routing zone, routes are determined on-demand (i.e. reactively), and it can use any on-demand routing protocol to determine a route to the required destination. Route creation is done using a query-reply system. The destination in turn sends back a reply message via the reverse path and creates the route.

Advantage:
- ZRP is since uses both reactive and proactive schemes, it exhibits better performance.
- Since hierarchical routing is used, the path to a destination may be suboptimal.
- Since each node has higher level topological information, memory requirement is greater.

III. Related works

A. Energy Aware OLSR (OLSR_EA)
The functional modules of OLSR needed to be modified to build OLSR_EA include the routing table computation algorithm and the TC message generation/reception processes(Guo et al, 2011)(Maleki et al, 2003) Also three important functions are needed to be added: the acquirement of current energy level of the node from MAC layer, the management of the adaptive energy consumption prediction, and the energy cost calculation and storage. The rest of OLSR are remained unchanged.

They developed a node-cost-based routing table computation algorithm called TierUp to replace the original hop-count based routing table computation algorithm of OLSR. TierUp is a modified version of Dijkstra's algorithm. The essential difference between TierUp algorithm and Dijkstra's algorithm is in the treatment of individual costs: while Dijkstra's algorithm evaluates individual costs on link basis, TierUp algorithm evaluates individual costs on node basis. The nature of shared radio channel of a wireless network diminishes the differences among the physical links. Thus those link costs affected by node attributes (e.g., delay) are only node specific. Their energy cost metric is certainly another node-specific attribute. In this sense, TierUp algorithm can be viewed as a node state (rather than a link state) routing table computation algorithm.

The other important difference between TierUp algorithm and Dijkstra’s algorithm is the reason for the name TierUp. In Dijkstra’s algorithm, the cost from the source node s to a node v may be
updated by the smaller cost through one of v’s neighbors, which may have shorter, same, or even longer hop distance to s compared to v’s hop distance to s. Contrastively, in TierUp algorithm, the cost from s to v will never be updated by any smaller cost through one of v’s neighbors which hop distance to s is longer than the hop distance from v to s. If we call the nodes having the same hop distance to the source node s as peer nodes in the same tier, the cost from s to any node in a certain tier isn’t used to update the cost from s to any node in any lower tier. There are three Target of this special treatment: to avoid spreading channel contention to bigger area due to taking route with longer hop distance, to reduce the probability of broken route due to going through more nodes, and to save end-to-end energy consumption.

Centralized by TierUp algorithm, there are a few other important functional modules dedicated to energy cost processing in OLSR_EA. The interconnections among these functional modules are depicted in Fig. 6, in which TC stands for Topology Control. The most recent certain number of 30 measured energy consumptions are sent to the ARIMA Prediction Facility module for either model fitting (training) or predicting. The model fitting is conducted regularly or whenever the prediction error threshold (0.2 in their implementation) is exceeded. The ARIMA Prediction Facility module is implemented as a static library.


```
Function TierUp (V, W, s)  # V is the aggregation of the nodes, W is the aggregation of the weights, s
q[0] = 0  # s is the source node
for each node v in V
    if (v ∈ V_in(s))
        q[v] = w_v
        previous[v] = s
    else
        q[v] = infinity
        previous[v] = undefined
end if
end for
F = {v, V_out(s)}  # F contains the nodes which shortest paths are known based on available cost info
Q = V - F  # Q is the opposite of F
Q = Q ∪ {f | f ∈ tier}
New = (null)
while Q is not an empty set
    Q = Q - New
    New = (null)
    for each c[v] in Q  # T contains the costs from s to the nodes which are 1 more hop away from s
        c [next hop[v]] = c[v] + w_v
    end for
    Ben(c[next hop])  # T contains the costs from s to nodes in the next tier
    for each v’s neighbor in the same tier as v
        if (c[v] ∈ Q)
            c[v] = c[next hop[v]]
            previous[v] = V
            F = F ∪ {v}
        else
            T = T ∪ {c[v]}
        end if
    end for
    T = T ∪ {q[v]}
    end if
end while
```

Within OLSR_EA there are four important user-configurable parameters related to energy consumption prediction and routing table computation. They are:

*arimaInitial_time*: the time when the measured energy consumption time series is first analyzed to find a fitting ARIMA model. It must be a time big enough to allow for sufficient (normally at least 15) number of available instances in the energy consumption time series to reveal significant statistical characteristic of the time series.

*arimaRetrain_ival*: the interval of regular ARIMA model fitting applied to the most recent certain number of values in the energy consumption time series. Because of those highly dynamic affecting factors of energy consumption in a MANET, they cannot trust a fitting ARIMA model all the time and we need to frequently update the ARIMA model corresponding to the most recent per-TC-interval energy consumptions.

*prediction_window*: the number of TC intervals to be covered by one time of prediction.

*ncosts_freshness*: the proportion of up-to-date energy costs in the total needed energy costs for routing table computation.

### A. Intelligent characteristics of OLSR_EA

Their energy-aware routing system, OLSR_EA, is an intelligent network system that adapts to the dynamic network energy consumption behaviors proactively. A generic network intelligence framework is proposed in (Toh et al,2001)(Moganti et al,2001), in which network intelligence is basically defined as the ability of a network to recognize and respond to the network condition changes so as to optimize the utilization of network resources and accomplish the goals of the service user and/or provider. In brief, the key features of an advanced intelligent network system include environment reorganization, information reasoning and assessment, prediction of future network condition, and autonomous networking control.

### B. OLSR-aware centralized channel access scheduling

In creating solutions for channel access scheduling, different forms of graph-coloring algorithms are widely used. Given an undirected graph $G = (N, L)$, vertex coloring is the assignment $\alpha : N$
→ C of colors (C) to vertices (N) such that no two adjacent vertices get the same color and the number of colors used is minimized (Malaguti et al, 2010)(Mecke et al, 2007). Finding the minimum number of colors in this assignment process is shown to be NP-hard (Rhee et al, 2004).

Since the slot assignment problem is NP-hard, there are several heuristics proposed to provide an approximate solution. Between these heuristics, FF\(^1\) and Degree-Based Ordering are among the best-known solutions (Omari et al, 2007).

In the FF Vertex Coloring Algorithm, there is a list associated with each color, holding the nodes that are already assigned to that color. When an unassigned node i is to be assigned a color, the FF Vertex Coloring Algorithm begins by checking the already assigned nodes list associated with each color and assigns the first suitable color j. A color j is called appropriate for node i if node i does not conflict with any of the nodes that are already assigned color j. A new color is allocated to node i if all colors used so far are unsuitable. In the FF Vertex Coloring Algorithm, no particular strategy is applied for the selection order of the nodes to be colored. On the other hand, in the Maximum Degree First (MDF) Vertex Coloring Algorithm, the vertex with the highest number of neighbors is selected first, providing an intuitively better coloring than the FF Vertex Coloring Algorithm (Omari et al, 2007). On the other hand, distance-d coloring is a special form of vertex coloring. In distance-d coloring, the colors are assigned such that no two vertices of distance d or less share the same color. TDMA channel access scheduling using the 2-hop interference model reduces to distance-2 coloring when the time slots are perceived as colors and both the primary and the secondary conflicts need to be avoided.

Their OLSR-aware centralized scheduling scheme (OA-C) uses a modification of the Distance-2 Maximum Degree First Vertex Coloring Algorithm for slot allocation. They argue that the size of the MPR Selector Set is a good predictor for the amount of traffic that can pass through a node, assuming that all nodes are eligible to generate traffic destined to any other node in the network (Kas et al, 2011). Therefore, in their OA-C scheme, they modify the Distance-2 Maximum Degree First Vertex Coloring Algorithm to integrate the MPR-based weighting scheme (see Algorithm).

As a result, Their OA-C algorithm has two major differences from the traditional Distance-2 Maximum Degree First Vertex Coloring Algorithm.

1. Each node i ∈ N is associated with an MPR-based weight Wi and it is assigned Wi time slots in a single scheduling cycle.
2. Nodes in N are sorted in a non-increasing order with respect to their MPR-based weights. In this way, the nodes whose assignments resolve more conflicts (both primary and secondary) are assigned first and the nodes that are assigned later are less likely to require new slots, resulting in a smaller scheduling cycle length.

\(^1\)First-Fit
Pseudo Code.2. OLSR-Aware Centralized Scheduling (Kas et al, 2011).

Data: Undirected graph $G = (N, L)$ where $N$ is the set of nodes and $L$ is the set of links connecting the nodes in $N$.
Data: $W$: Weight vector.
Result: Each node in $i$ in $N$ is assigned $W_i$ many slots such that no two nodes within the same 2-hop neighborhood are assigned the same slots.

1. $N \leftarrow \text{Sort} (N, W, \text{Nonincreasing});$
2. $\text{cycle\_count} \leftarrow 0;$
3. for $i \leftarrow 1$ to $|N|$ do
   1. $j \leftarrow 1;$
   2. while $j < W_i$ do
      1. $\text{count} \leftarrow 0;$
      2. while $\text{IsFeasible} (i, \text{count}) = \text{FALSE}$ do
         1. $\text{count} \leftarrow \text{count} + 1;$
         2. $\text{slots}[\text{count}].\text{Add}(i);$  
      3. if $\text{count} > \text{cycle\_count}$ then
         1. $\text{cycle\_count} \leftarrow \text{count};$
      4. end
   3. $j \leftarrow j + 1;$
   4. end
4. end

In Algorithm, they make use of a sub function called IsFeasible. The IsFeasible function ensures that no other nodes within the 2-hop neighborhood of the given node $n_id$ are scheduled to transmit at the given time slot slot_number. In the function, $N1.n_id$ and $N2.n_id$ represent the 1-hop and 2-hop neighbors of node $n_id$, respectively.

C. Multi-objective OLSR protocol

In this work they integrated mechanism of metric prediction and evaluation with OLSR, (OLSR_MO) (Guo et al, 2011). The routing table calculation is derived from Dijkstra’s algorithm with the multi-metric composite cost representing the costs in the algorithm. The essential difference between MTierUp algorithm and Dijkstra’s algorithm lies on the treatment of individual costs. While Dijkstra’s algorithm evaluates individual costs on link basis, MTierUp evaluates individual costs on both node basis (for queuing delay and energy cost) and link basis (for link stability cost). Moreover, in Dijkstra’s algorithm, the cost from the source node $s$ to a node $v$ may be updated by the smaller cost through one of $v$’s neighbors, which may have shorter, same, or even longer hop distance to $s$ compared to $v$’s hop distance to $s$. But, in MTierUp, the cost from $s$ to $v$ will never be updated by any smaller cost through one of $v$’s neighbors where hop distance to $s$ is longer than the hop distance from $v$ to $s$. If we call the nodes having the same hop distance to the source node $s$ as peer nodes in the same tier, the cost from $s$ to any node in a certain tier is not used to update the cost from $s$ to any node in any lower tier. There are three reasons for this special treatment: avoid spreading channel contention to bigger area by taking route with longer hop distance, to reduce the probability of broken route by going through more nodes, and to save end-to-end energy consumption.

Besides MTierUp routing table calculation algorithm, there are a few other important functional modules dedicated to multi-objective metric processing in OLSR_MO. The flowchart of these
functional modules is depicted in Fig. 7, in which DES stands for Double Exponential Smoothing, and TC stands for Topology Control. The most recent certain number of measured mean delays and energy consumptions are sent to the DES parameter estimation module only in two cases: The initial state when the two parameters (alpha and beta) of DES are to be decided for the first time and the time when the prediction error threshold is exceeded and the two parameters are needed to be re-estimated.

![Figure 7: The structure of the multiple routing metrics processing subsystem in OLSR_MO](Image)

The DES parameter estimation module is implemented as a static library utilizing levmar (Guo et al, 2011), which is a C implementation of LMA. After at least 7 measured values are collected for both the mean queuing delay time series and the energy consumption time series (recall that seven measured values are needed to estimate the DES parameters a and b), future mean queuing delays and energy consumptions are continuously predicted using DES. They are then normalized into relative delay cost (integer between 0 and 255) and composite energy cost (integer between 0 and 255) before being saved locally and broadcasted to other nodes. The link-sensing module and the TC message processing module track the link ages of local links and remote links respectively, the relative link stability costs are then calculated based on the link ages. The TC message is extended to carry the predicted delay costs and energy costs. If the MANET is heterogeneous in terms of node power consumptions and/or node speed, the TC message is also used to carry the transmission range and moving speed of the originating node for the calculation of Lpd. Since only the most recent predicted delay and energy are sent out, the total length taken by the delay costs and energy costs in a TC message is 2S(pw) bytes, where S(pw) is the size (number of covered intervals) of the prediction window. The starting time of this time-span is indicated by another field in a TC message, whilst the ending time of is simply the starting time plus the length of the prediction window. They make this field occupy the two
bytes reserved in the original OLSR TC message. Therefore no extra messaging overhead is imposed by introducing this new field. The resulting TC message format is shown in Fig. 8. Since OLSR encapsulates multiple control messages in a UDP packet, additional header lengths need to be accounted in the total packet length. These additional header lengths include 4 bytes of OLSR packet header, 8 bytes of the UDP header, 20 bytes of the IP header, and 34 bytes of the MAC 802.11 header. Considering an OLSR control packet encapsulating only m TC messages, its total length is then $\sum_{k=1}^{m}(16 + 4N_k)$ bytes. The extra overhead of using OLSR_MO TC message format in such a control packet is then $S(pw) _ 2m$. Considering that $S(pw)$ is normally small (e.g., 2), such an extra overhead is relatively light compared to the total packet length. In addition to the TC message (s), if at least a HELLO message or a MID message is encapsulated in an OLSR_MO control packet, the extra overhead of using OLSR_MO TC message format is even lighter.

Figure 8. the extended TC message format of OLSR_MO (Guo et al., 2011).

D. Multipath OLSR (MP-OLSR)

Method MP-OLSR is proposed to enhance load-balancing, energy-conservation, QoS and security. MP-OLSR is a hybrid multipath routing protocol that takes advantage of the MPR mechanism to flood the network with control traffic information (Yi et al., 2011). In MP-OLSR, the OLSR proactive behavior is changed for on-demand route calculation. MP-OLSR protocol becomes a source routing protocol. There are two stages: topology discovery and routes calculation. During the topology discovery phase, nodes obtain a partial topology map just like in OLSR. However, MP-OLSR nodes do not construct routing tables. During the routes computation phase, nodes calculate multiple paths to reach any other node in the network following on-demand scheme. MP-OLSR implements Multiple Description Coding (MDC) for data transfer. MDC adds redundancy to information streams and divide them up into several sub-streams to improve the integrity of data. These sub-streams are sent along several paths from the source to the destination. MP-OLSR implements source routing with route recovery and loop detection to adapt to the changes in the network topology. Therefore, when data transfer is required, route recovery and loop detection allow every node to detect if a path is not valid anymore and to find a new path to reach the final destination. MP-OLSR implements the Multi Path Dijkstra’s algorithm to discover the shortest routes. The paths that are obtained can be grouped into two categories:

1. Disjoint: In this category they have two types of disjoint paths: node-disjoint and link-disjoint. Node-disjoint paths type does not share nodes except for the source and destination nodes. Link-disjoint paths can share some nodes but all the links are different.

2. Inter-twisted: In this case, the paths may share several links.
To construct disjoint paths, MP-OLSR defines cost functions to obtain new paths that tend to be node-disjoint or link-disjoint. Once a path is computed, a function $f_p$ is used to increase the cost of the links that belong to the computed path, e.g., $f_p(c) = 3c$. A function $f_c$ is defined to increase the cost of the links of the nodes included in the path previously obtained. In MP-OLSR, neither nodes nor links used in computed paths are eliminated. This strategy allows MP-OLSR to construct multiple paths in sparse networks where it is not always possible to find node-disjoint paths. Additionally, to increase the chances of constructing node-disjoint paths, the MPRs report all their one-hop neighbors (i.e., the TCR parameter is equal to two). Consider $f_{id}$ as the identity function, i.e., $f_{id}(c) = c$. Therefore, to construct disjoint paths, there are three possibilities:

1. If $f_{id} = f_c = f_p$, then paths tend to be link-disjoint.
2. If $f_{id} < f_c = f_p$, then paths tend to be node-disjoint.
3. If $f_{id} < f_c < f_p$, then paths tend to be node-disjoint, but when not possible they tend to be link-disjoint.

### Route computation

The goal of MP-OLSR is to construct a set $P$ of $t$ paths, with no loops, between a source node $s$ and a destination node $d$ (Zhou et al., 2005) (Cervera et al., 2013). The network topology is represented by a graph $G = (V, E, c)$, where $V$ is the set of vertices $v$ (i.e., nodes), $E \subseteq V \times V$ are the set of arcs $e$ (i.e., links between nodes) and $c$ a strictly positive cost function. An arc $e \in E$ is defined as a pair of vertex $(v_q, v_{q+1})$ with a bidirectional link. A path between a pair of distinct vertices $(s, d)$ is defined as a sequence of vertices $(v_1, v_2, \ldots, v_m)$ such that $(v_q, v_{q+1}) \in E$, $v_1 = s$ and $v_m = d$. The cost of an arc formed by the vertices $v_q$ and $v_{q+1}$ is noted $c(e_q) = c(v_q, v_{q+1})$ and is always positive.

Pseudo Code.3. Function. Multi path Dijkstra $(s, d, G, t) \Rightarrow P$ (Yi et al., 2011).
MP-OLSR uses the hop count as the metric to select multiple paths, none the less, other metrics can be used to select the best path, e.g., bandwidth, delay, etc. Given a source nodes, MP-OLSR will keep an updated flag (updated Flag) to identify if the routes are still valid.

The advantages of a link state multipath approach are clearly exemplified. Classical issues in MANET are covered: scalability, lifetime of the network (by reducing the number of forwarded packets per node) and non reliable wireless transmissions.

E. Smart Prediction

As energy level is a monotonically decreasing metric, a node can adjust “old” information and predict the current value at any time it needs to determine the residual energy level for a visible node (when selecting MPRs or determining a routing path, for example). Their idea is therefore to have every node locally adjust nodes’ old energy levels based on their past energy consumption rate. They have proposed a Prediction mechanism in which each node locally extrapolates the expected energy level based on old (reported) energy levels and the energy consumption rate for that node based on the most recent two reported values. For example, in their simulation results are obtained: at second 51 of the simulation, node 0 has an energy level of 958.581 J associated with node 1 and this knowledge is time stamped 47.5916 s. At second 52, the perceived energy level for node 1 (from the perspective of node 0) is 954.998 J, time stamped 49.5884s (Kunz et al., 2007). Node 0 computes the consumption rate of node 1 as:

\[
\frac{(958.581 \text{ J} - 954.998 \text{ J})}{(49.5884 \text{ s} - 47.5916 \text{ s})} = 1.7943 \text{ J/s}.
\]

To determine the perceived residual energy level for node 1 at second 52, node 0 adjusts the last reported energy level of node 1 by the consumption rate and elapsed time: Predicted perceived residual energy of node 1:

\[
954.998 \text{ J} - (1.7943 \text{ J/s} \times (52 \text{ s} - 49.5884 \text{ s})) = 950.6707 \text{ J}.
\]

If no prediction is possible, as no consumption rate is known yet, the last reported energy level will be used without adjustment.

A disadvantage of the Prediction algorithm is the need to wait for two different perceived value readings, Thus the a consumption rate can be calculated and used to adjust the perceived values. These results capture how often they are successful in adjusting the last reported residual energy level as a percentage of the number of times they need knowledge about a node’s residual energy level (to select MPRs, determine routes, or report perceived energy levels for the purpose of collecting statistic). Under high traffic loads, adjustments happen less rarely. Protocol control messages are lost and delayed, and consequently the nodes won’t “hear” the other nodes. Following a node is deemed inaccessible; they go through the startup phase again, where they need at least two successive reports to be able to calculate a consumption rate. They therefore
propose the Smart Prediction algorithm which is an enhanced version of the prediction algorithm so that adjustments take place almost all the time (Kunz et al., 2010).

In the Smart Prediction algorithm, for every pair of nodes (n1, n2), if n2’s consumption rate is not yet known, n1 adjusts the perceived value of n2’s residual energy level based on the average of all known consumption rates for other nodes. If n1 knows not a single consumption rate for other nodes, it adjusts n2’s perceived energy level based on its (n1’s) consumption rate. Utilizing all known nodes’ consumption rates eliminates the domination of outliers and ensures the closeness to the actual consumption rate; we can assume that nodes are somewhat homogeneous in the energy characteristics of their wireless cards.

Due to the Smart Prediction algorithm addresses the problem of not being able to adjust the perceived energy level value at all times, it achieves much better performance in terms of overall inaccuracy level, particularly under higher traffic rate. The Prediction and the Smart Prediction algorithms better the Default OLSR protocol. While at the same time, the Smart Prediction algorithm outperforms the Prediction algorithm in improving the overall inaccuracy level.

### F. Energy-Efficient OLSR (EE-OLSR)

De Rango et al. have proposed a heuristic to set the willingness value in OLSR Protocol. Each node according to its residual battery and predicted lifetime decides to attribute a low (WILL_LOW), a default (WILL_DEFAULT) or a high (WILL_HIGH) value to its willingness variable. The proposed heuristic is depicted in Fig. 12. Note that the predicted lifetime in seconds at time step t, $\text{Lifetime}$, is calculated as in Eq. 1:

$$\text{Lifetime} = \frac{\text{RE}_t}{\text{DRate}_t} \quad (1)$$

$\text{RE}_t$ and $\text{DRate}_t$ denote, respectively, the residual energy and the energy drain rate at time step $t$. $\text{DRate}_t$ is computed using the exponential moving average method as proposed in (Kim et al., 2002). To measure the energy drain rate per second, each node monitors its energy consumption during a $T$ seconds sampling interval.

$$\text{DRate}_t = \alpha \times \text{DRate}_{t-1} + (1- \alpha) \times \text{DRate}_{\text{sample}} \quad (2)$$

$\text{DRate}_{t-1}$ indicates the drain rate calculated in the previous interval, whereas $\text{DRate}_{\text{sample}}$ is the newly observed energy drain rate value.
In EE-OLSR implementation, less than 10% of residual battery capacity is considered as low value. In addition, less than 10 seconds of predicted lifetime is considered to be short, whereas more than 100 seconds is long lifetime. The authors claim that their willingness setting policy contributes in a better load balancing where low batterychargednodes are avoided in comparison to the standard OLSR.

G. The Proposed Fuzzy-based Energy Aware OLSR (FEA-OLSR)

To compute the willingness parameter, in FEA-OLSR, each node uses a FLS (CHETTIBI et al, 2013). In this latter, the Remaining Energy, RE, and the Expected Residual Lifetime, ERL, are the FLS inputs. The linguistic terms used to qualify them are: “Low” and “High”. Note that all the input membership functions are Trapezoidal. A Trapezoidal membership function, $\mu(x)$, is defined by Eq.3:

\[
\mu(x) = \begin{cases} 
\frac{x-a}{b-a} & \text{if } x \in [a,b] \\
\frac{1}{b-a} & \text{if } x \in [b,c] \\
\frac{d-x}{d-c} & \text{if } x \in [c,d] \\
0 & \text{otherwise}
\end{cases}
\]  

(3)

The output of the fuzzy logic system is the node willingness to be chosen as an MPR node. To qualify the output, the terms “WILL_Low”, “WILL_Default” and “WILL_High” are used.

The inference engine for the fuzzy system follows a Zero order Sugeno fuzzy model. The Sugeno fuzzy model (also known as TSK model) was proposed by Takagi, Sugeno and Kang in
A typical inference rule in a Sugeno fuzzy model has the following form: if input1 = x and input2 = y then output z = ax + by + c

For a zero-order Sugeno model, the output z is a constant (i.e. a=b=0). The output z_i of each rule is weighted by the firing strength w_i of the rule. The final output of the system is the weighted average of all rule outputs, computed as shown in Eq. 4. N denotes the number of fuzzy rules. The proposed fuzzy rules base is introduced in Table 1.

\[
Final \ output = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i}
\]  \hspace{1cm} (4)

Where:

\[
w_i = MIN(\mu(RE), \mu(ERL))
\]  \hspace{1cm} (5)

Table 1: Fuzzy Rules Base (CHETTIBI et al, 2013).

<table>
<thead>
<tr>
<th>FLS Inputs</th>
<th>FLS Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE</td>
<td>ERL</td>
</tr>
<tr>
<td>LOW</td>
<td>LOW</td>
</tr>
<tr>
<td>LOW</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>LOW</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Our review of the results of the simulations show EE-OLSR and FEA-OLSR belong to the Maximum-Lifetime routing family. Therefore, their main objective is to extend network lifetime and FEA-OLSR has marked the lowest delay in both low and high traffic scenarios. In fact, this is a suitable feature for real-time applications that require a short end-to-end delay.

IV. Summarize of improvements on the protocol OLSR

<table>
<thead>
<tr>
<th>Method</th>
<th>Routing protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLSR_EA (Maleki et al, 2003)</td>
<td>Energy Aware OLSR</td>
<td>OLSR_EA, an extended version of OLSR with integration of their energy preserving mechanism, is able to prolong network</td>
</tr>
<tr>
<td>Scheme</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>OA-C (Kas et al, 2011)</strong></td>
<td>OLSR-aware centralized channel access scheduling</td>
<td>OA-C scheme, modify the Distance-2 Maximum Degree First Vertex Coloring Algorithm to integrate the MPR-based weighting scheme. OA-C algorithm has two major differences from the traditional Distance-2 Maximum Degree First Vertex Coloring Algorithm.</td>
</tr>
<tr>
<td><strong>OLSR-MO (Guo et al, 2011)</strong></td>
<td>Multi-objective OLSR protocol</td>
<td>OLSR_MO developed an algorithm that extends the Dijkstra’s algorithm to handle the multiple objective routing problems using a composite utility approach.</td>
</tr>
<tr>
<td><strong>MP-OLSR (Yi et al, 2011)</strong></td>
<td>Multipath OLSR</td>
<td>(MP-OLSR) is the multipath optimized link state routing. The extension of the single-path version includes a major modification of the Dijkstra algorithm (two cost functions are now used to produce multiple disjoint or non-disjoint paths), auxiliary functions, i.e. route recovery and loop detection to guarantee quality of service and a possible backward compatibility based on IP-source routing.</td>
</tr>
</tbody>
</table>
V. Conclusion

In this survey we reviewed spent all kinds of routing protocols in MANET, and some of them explained then work on the protocol discussed OLSR and improvements. In summary it can be said that, in OLSR_EA, our reviews showed that OLSR_EA improves the level of traffic balance among the nodes and therefore increases packet delivery ratio when node mobility is low. However, routing metrics other than energy cost have to be considered to improve packet delivery ratio when node speed gets higher. In OLSR-aware centralized channel access scheduling, our reviews showed that OA-C performs better than the other schemes in terms of the number of delivered/dropped packets and the packet delivery ratio. OLSR_MO is shown to be an effective approach for finding multi-objective solutions for network problems in terms of minimizing data packet delivery ratio, shorter average end-to-end delay, spending less energy, and prolonging the network lifetime. The MP-OLSR could effectively improve the performance of the network (especially in the scenarios with high mobility and heavy network load) and also be compatible with OLSR. Under the Prediction technique, a node’s energy level is adjusted based on its past behavior (its own consumption rate). Smart prediction is a modified version of the Prediction technique such that, if no consumption rate can be determined for a node, its energy level is adjusted based on the average of all known consumption rates for other nodes. Our reviews showed that both
Prediction and Smart Prediction outperform the Default OLSR. Moreover, Smart Prediction outperforms Prediction since energy level adjustments take place all the time. The Our reviews showed that EE-OLSR outperforms OLSR in terms of throughput, average nodes lifetime, connection expiration time, nodes lifetime, preserving the normalized control overhead. And the last action, they addressed the problem of OLSR willingness-parameter setting according to nodes energy-profiles using a Fuzzy Logic Zero-Order Sugeno system. Our reviews have confirmed the out performance of FEA-OLSR in comparison to EEOLSR an energy-aware heuristic based variant of OLSR.

References


